# Explicit Application-Network Cross-layer Optimisation

Dimitrios P. Pezaros, Laurent Mathy

*Computing Department, Lancaster University*
*Infolab21, UK*
{dp, laurent}@comp.lancs.ac.uk

***Abstract*— The emergence of overlay network applications that rely on application-level decisions for many aspects of their operations (e.g. routing, content replication, etc) creates cross-layer interaction issues with ISP network operations. Indeed, the independent optimisation of a diverse set of objectives using layer-local information can lead to operational instability and sub-optimal resource usage. We argue that an explicit interaction between the application and network layers can provide benefits for each layer. We postulate that such cross-layer interaction must however fulfil two conditions to be pragmatic and acceptable: 1) no explicit information about the structure and operations of each layer must be ex-changed; 2) each layer must be able to independently set its own policies and objectives. Because we limit this interaction to application hosts and their access ISP, the proposed method is also incrementally deployable. We show, through evaluation of simple examples, that explicit cross-layer interaction does indeed bring performance benefits to all parties, for applications ranging from simple client-server to more complex overlay network scenarios.**

## I. Introduction

The increasing popularity of overlay networks to deploy customisable and reliable services at the application layer by implicitly or explicitly taking control over routing can lead to sudden, highly variant and unpredictable traffic dynamics over the underlying Internet infrastructure. File sharing, application-level multicast, scalable object location and network-embedded storage are only a few examples of such overlay services. Depending on the target application domain, overlays employ their own internal mechanisms and routing strategy to optimise certain aspects of their performance. Having multiple overlays simultaneously operating over segments of the Internet, with each one independently and dynamically deciding how to route traffic, can be cost-ineffective, and also work against traffic engineering and load balancing policies adopted by ISPs at the underlay network layer [10][8]. The root of the problem is simply that both the overlay and the underlay operate in an independent manner. More fundamentally, layered network design has almost imposed a form of segregation in the decision making process at each layer. For instance, a network is often engineered and provisioned considering matrices of traffic aggregates at large time-scales that are very coarse-grained compared to overlay reaction times. On the other hand, applications, and overlay networks in particular, mostly consider the underlying network as a black box. The application layer then routinely probes the underlying network to solve tasks such as proximity estimation. Such probing has been observed to create "ping storms" [11].

In order to address the scalability, overhead and stability issues of such globally-uncoordinated actions, it has been suggested that a routing underlay service should reside above the underlying Internet and expose global topology and/or performance information to assist applications and overlay networks in their operations [11]. However, exposing topology and/or any other operational network characteristics to the application layer may indeed prove impractical. For a start, many ISPs may be reluctant to export such information which they often treat as trade secrets and competitive advantage sensitive. Furthermore, exporting up-to-date and global information may not be scalable and contradict ISP's local policies. For instance, exporting global topology information not only may not scale to a net-work the size of the Internet, but can also give applications the opportunity to by-pass ISP routing policies. Another approach to solve the layer-interaction issues is to push support for the applications into the network. This approach is classically exemplified by caching. However, the proliferation of applications and overlay networks also means a proliferation of cache types, a proposition that may not be very attractive for ISPs, especially be-cause caching can be a Digital Right Management liability and is often powerless in the face of end-to-end encryption.

In this paper, we postulate and demonstrate that some application-network cross layer issues can be addressed by an explicit interaction between the two layers, but without explicit exchange of structural information. We envisage that ISPs could deploy services that would take some (explicit) input from applications and explicitly return hints about these inputs. Applications would then make use of these hints to try and improve their operation and perceived performance, while hopefully, the use of network hints by applications would also improve performance at the network layer. An important point here is that how inputs are treated to form hints, and how hints are used is entirely up to the respective layers: each layer's prerogative to set its own policies, and optimize its own performance metrics, should be preserved. The remainder of the paper is structured as follows. Section II de-scribes an ISP hint service based on the idea of input clustering and section III outlines a number of applications using ISP hints. Section IV shows, through simulations, how simple ISP hint services can be exploited by various applications to provide cross-layer optimization. Section V discusses our findings and concludes the paper.

## II. ISP HINTS

ISP hints are a very general service offered by ISPs to local hosts and applications. We envisage that it takes the form of a request-response service between hosts and "hint servers" inside the ISP (access) network. The idea behind ISP hint services is very generic in the sense that the hint services are open and non prescriptive. An ISP could basically offer a range of such services, each taking specific inputs and returning specific hints. Applications would choose the hint service(s) that best fit their needs. As already pointed out, a salient feature behind the ISP hint ser-vice idea is that all entities involved can choose policies and performance optimization targets as they see fit. That is, which hints to choose and how to use them is up to the application, while which hint services to offer and how hints are computed is up to ISP. The goal is of course for all parties to gain from the use of ISP hints. However, ISPs could take measures to incite or enforce use of hints (such measures are outside the scope of this paper).

To fix ideas, we now describe two simple examples of ISP hint services, which we use in the remainder of the paper. The first such ISP hint service is called a "distance service". It simply takes IP addresses as input and returns a distance measurement between each destination and the requester. The notion of distance can obviously be specified and measured in numerous ways, but again the advantage of using the ISP hints abstraction is that ISPs can choose which metric to provide and how to measure it, without even having to inform the applications (the requesters) of their choices. For instance, an ISP may decide to use Autonomous System (AS) path length as a distance measurement, while another may base its distance measurements on more complex embedded coordinate systems [13][12]. The "region-aware clustering" service is another example of ISP hint service. Here, the ISP would take a set of IP addresses as input, and return these addresses split into several subsets (clusters). In its simple version, all the addresses in the same subset/cluster would be reached, from this ISP, through the same egress border router. However, note again, that ISPs may want define "regions" in a different way. For example, the clusters in the simple region-aware clustering could be further split according to the second AS hop they would traverse, and so on and so forth. This latter version of the service could be called "2-level hierarchical region-aware clustering". Note that the distance service based on AS path length, and both region-aware services described above can be implemented based solely on routing information available at an ISP. Indeed, all the information required can be extracted from a BGP Routing Information Base [17] and the corresponding hint services could therefore readily be realized as an extension to BGP route servers.

## III. APPLICATIONS USING ISP HINTS

In this section, we illustrate how applications can use ISP hints to improve their perceived performance. We limit our discussion to the content distribution in the form of file transfers. The *client-server* paradigm is very often used for file transfers (e.g. HTTP, FTP). In the simplest form of client-server transfers, the server receives requests from the clients which it serves. The server usually accepts new requests as long as it has enough local resources to accommodate them. On the other hand, an enhanced server could exploit ISP hints to increase the perceived service quality of its clients. Indeed, by using the simple region-aware clustering service, a server is capable of ensuring some form of load-balancing between its ISP egress points, thus controlling the contention that exists between its clients inside the network. To do so, the server could set a limit on the number of con-current connections that it will open per cluster (i.e. ISP egress router), and use the simple region-aware clustering for admission control. More specifically, the server can use the ISP clustering service to decide which request to serve (if the corresponding connection uses an egress for which the connection limit, a.k.a. threshold, has not been reached) or queue. By limiting the number of concurrent downloads and ensuring load-balancing, the average download time should be reduced, improving user satisfaction. We have implemented two client-server (CS) algorithms and quantified the performance gains of employing ISP hints services. A traditional CS algorithm has been developed where one node serves simultaneous requests up to a certain threshold value $T$, on an aggregate First-Come-First-Served (FCFS) basis. Upon reaching this threshold, further incoming requests are queued and served as soon as existing transfers complete on a FCFS basis. An enhanced CS algorithm has also been developed where the server uses ISP hints to perform load-balancing on the in-coming client requests. As with traditional CS case, up to a threshold $T$ client requests are served simultaneously. However, in this case, the server uses the region-awareness ISP hint to cluster incoming requests based on the egress links response traffic is going to be routed through. Traffic is then load-balanced over the server's egress links. Three variants of this region-aware request clustering have been implemented. In the Simple Clustering (SC) variant, the server clusters requests to regions based on the first-hop egress link traversed by the response traffic. The total simultaneous request threshold $T$ is divided by the servers' $v$ egress links and $\tau = T / v$ simultaneous requests are served per-cluster. In the second-hop Flat Clustering (FC) variant, the server groups requests based on the second hop traversed by the response traffic, and performs load-balancing ignoring first hop information. Finally, in the second-hop Hierarchical Clustering (HC) variant, requests are clustered hierarchically based on the first and second hops traversed by response traffic. The initial threshold $T$ is divided by the number of first-hop egress links $v$ to produce $i$ first-hop thresholds $\tau_i = T / v$, each of which is further divided by the number of egress links attached to first hop $i$. In all variants, when the region-aware clusters are computed, requests are served on an intra-cluster FCFS basis.

Another popular way to transfer files is through a file sharing Peer-to-Peer (P2P) *overlay* [6][9][14][5]. P2P file sharing comes in many flavours, so for this paper we chose to use a simplified version of a Bittorent-like distribution overlay as a reference [2]. A generic simple overlay algorithm has been implemented that includes two different peer entities, a

central tracker of the content; and normal peers acting as both content clients and providers. Each peer registers with the tracker either as a provider or as a requester for a certain piece of content. In the former case, the tracker updates its list of providers, and in the latter case it acknowledges the client's registration by sending back the list of currently available providers for the specific content. If the list is too large, then a number of providers is chosen randomly and returned to the requester. The client randomly selects a provider from the list to request the content from. If it is denied service (because the provider has reached the maximum number of connections it will serve), it randomly selects a different provider from its list until either its request is accepted, or its providers' list is exhausted, in which case it times out and re-requests a providers' list from the tracker after a certain time interval. Upon successful completion of a download, a peer registers itself as a provider with the tracker.

We have developed a number of augmented and region-aware overlay algorithm variants to demonstrate how the operation of this simple overlay can be optimised using explicit cross-layer interaction through combinations of the two ISP hint services described in section II. The augmented overlay algorithm uses the ISP "region-aware clustering" hint (as in the CS case described above) for the provider peers to load-balance their response traffic. At the same time, a provider that has reached its simultaneous serving threshold explicitly redirects further clients to the subset of peers (providers) it has already served through the same "regional" cluster that the incoming request came from. This way, providers attempt to spread the overlay traffic load to diverse segments of the underlay (Internet) infrastructure. Requesting peers choose among alternative providers either using a Random function (Ran), or by employing the ISP "distance" hint that returns an AS Proximity (ASP) metric and then selecting the least-AS-distant provider. Furthermore, a Region-aware Overlay (RegO) algorithm has been developed which implements Random (Ran) provider selection and region-aware load-balancing to requests. In contrast to the augmented algorithm where clients are only redirected to a "region"-based subset of the alternative providing peers, RegO uses the central tracker entity that provides requesters with all currently providing overlay peers. All variations of the augmented and the RegO algorithms have been designed to implement first-hop (SC) and second-hop hierarchical (HC) region-aware clustering to load-balance competing requests.

In the following section, we evaluate and quantify through simulation the performance gains of employing explicit cross-layer synergy for both the underlay and the overlay layers.

## IV. Evaluation

### A. Client-Server Load-Balancing

We have used the Network Simulator (ns-2) [18] to assess the effect of the region-aware load-balancing for the client-server case over a variety of Internet-wide topologies. A piece of content hosted by a single peer on the topology is simultaneously fetched by a number of clients (assumed to be triggered by an external stimulus, such as central web-based advertisement). The server has a configurable threshold up to which it is willing to serve client peers. The source of the requested content resides behind a number of egress links which we varied in different experimental runs. We have employed $10^3$ client nodes using both symmetric and random node placement over Internet-wide AS topologies. Symmetric client placement assumes the same number of clients attached to each AS. Under random client placement, clients are randomly attached behind each AS using a multi-level number generator based on uniformly distributed random variables. We have conducted multiple simulation runs over different edge-degree topologies and using a varying threshold value of 5, 10 and 20 simultaneous transfers, to compare the mean individual transfer throughput between the region-aware load-balanced scenarios and the unbalanced aggregate FCFS case.

Figure 1 shows the percentage increase in throughput for the different simultaneous transfer threshold values over symmetric client node placement. Throughput has been measured as the number of bytes transferred between the server and the client over the duration of the flow and resembles the widely used Bulk Transfer Capacity (BTC) metric. The link bandwidths of each topology generated were uniformly distributed to accommodate for the randomness in the actual available bandwidths over the Internet due to the variable traffic dynamics of each segment. Figure 2 shows the percentage increase of transfer throughput over random client node placement using the same threshold values and topology-wide link bandwidth distributions. It is evident that significant increase in transfer throughput is achieved by the cluster-based load-balancing algorithms with respect to the unbalanced aggregate FCFS mode of operation.

Over symmetric client placement, even first-hop SC load-balancing can achieve a steady over 10% mean throughput increase as the simultaneous transfers and access edge degree grow larger. Load-balancing based on second-hop FC and HC achieve throughput gains of up to above 20%. First-hop (SC) and second-hop hierarchical clustering (HC) show a steady and proportionally increasing trend with the number of simultaneous transfers and access edge degrees. Second-hop Flat Clustering (FC) exhibits a less predictable performance gain due to equalising (in some cases) the per-egress (first-hop) load balancing, however, for increasing numbers of simultaneous transfers this algorithm also shows a steadily anodic throughput gain. As expected, over random client placement, the throughput gain is influenced by the uneven number of clients accessed through each egress link, and is henceforth less deterministic. However, it is worth mentioning that throughput increase is still achieved by region-aware load-balancing in all cases, and also that absolute throughput values are in many cases larger than those of the corresponding threshold/edge degree values over symmetric client placement.

Overall, it is evident that an ISP can spread popular content faster to diverse segments of the Internet and minimise the persistence of incoming requests by employing a simple load-balancing algorithm and information readily available within its BGP speakers.

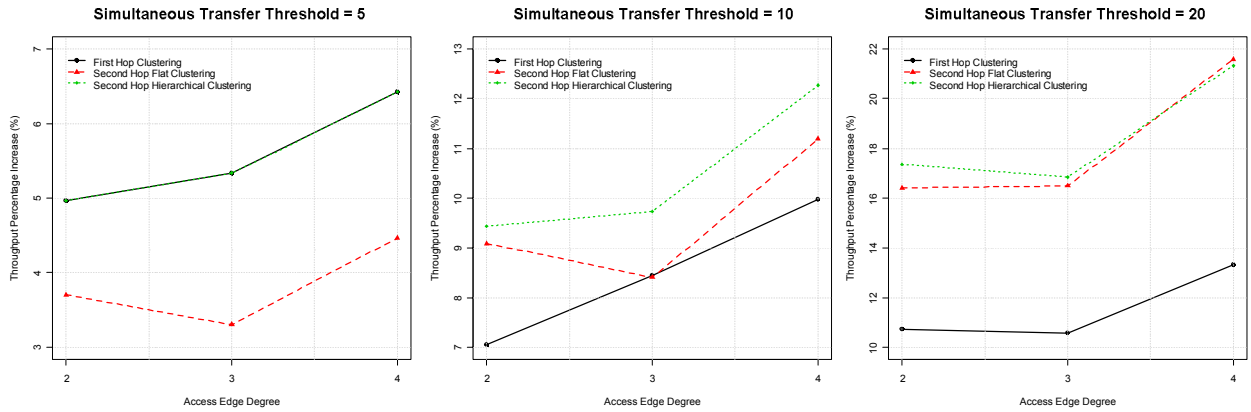


Fig. 1 Throughput percentage increase of load-balancing client requests under symmetric node placement for varying simultaneous transfer threshold values.


Fig. 2 Throughput percentage increase of load-balancing client requests under random node placement for varying simultaneous transfer threshold values.

### *B. Region-aware Overlay Algorithms*

We have further investigated the gains of region-aware load balancing for complete P2P file-sharing overlays over representative AS-level Internet topologies. We have hypothesised that region-aware load balancing by every providing peer will help the content to spread quickly over diverse segments of the underlying infrastructure, and hence minimise the impact of persistent request and response traffic on a single ISP that (implicitly) hosts the popular content. At the same time, the ISP itself does not need to know anything about the internals of the content other than to identify which requests are for the same highly-popular object (through e.g. hashes), nor does it need to invest into infrastructural support for services such as caching to minimise the additional stress over its links. We have evaluated the overlay/underlay interaction through comparative performance analysis of the different algorithms. Brite topology generator [3] was used to generate a large variety of representative topologies to include diverse topology models, AS node populations and minimum AS edge-degrees. We have generated a number of power-law AS-level topologies to include 100, 500, and 1000 nodes, each with a minimum edge degree of 2, 3 and 4 links per leaf AS [1][4]. Each simulation focused on both the initial bursty phase of the overlay when all peers simultaneously fetch a newly populated piece of content, and the steady-state phase of the system when content has spread among different peers. The experiments assessed the overall performance gains of the explicit underlay/overlay synergy in spreading the so-called first chunk of content among the participating peers [15][16]. The chunk size was set to 1MB. A constant threshold of 10 simultaneous transfers has been used. The performance metrics measured for all the algorithms and their variants were the individual transfer throughput in KB/s, and the mean and maximum link stress over the complete Internet-wide topology. As in the previous section, throughput is the BTC of each transfer, whereas mean and maximum link stress have been measured as the average and maximum number of flows active over each link of the topology, at any given time. The comparative results show that the region-aware algorithms improve both aspects of overlay and underlay performance.

Figure 3 shows the percentage improvement in mean transfer throughput over different-size AS-level topologies, achieved by the cross-layer algorithms with respect to theit pure overlay counterpart. The figure also shows the variations in throughput increase with respect to the minimum access edge degree of the topologies. Solid lines show the performance gains of the cross-layer algorithms with first-hop simple clustering (SC) and dashed lines show their second-hop hierarchical clustering (HC) counterparts.

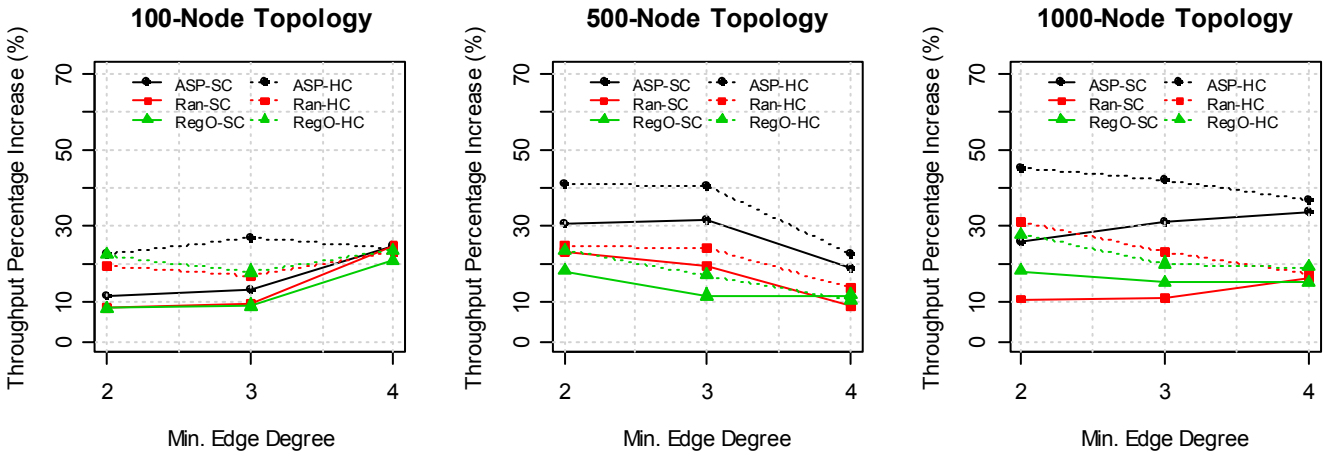


Fig. 3 Percentage increase in mean individual transfer throughput for the cross-layer algorithms

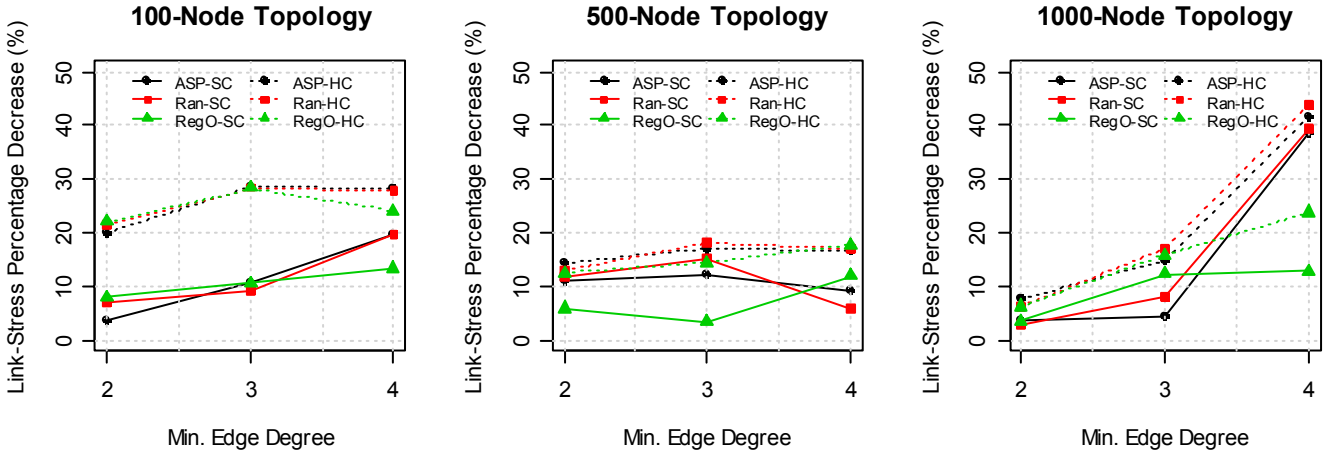


Fig. 4 Percentage decrease in topology-wide mean link stress for the cross-layer algorithms

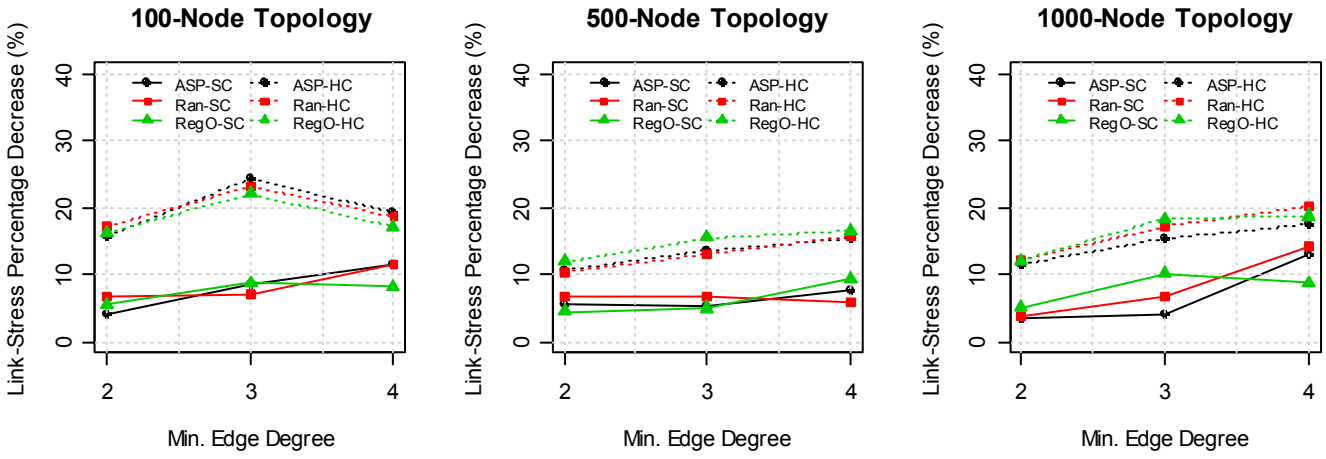


Fig.5 Percentage decrease in topology-wide maximum link stress for the cross-layer algorithms

Likewise, figure 4 and figure 5 show the percentage improvement in the topology-wide mean and maximum link stress for the same set of AS-level topologies. Although there seem to be no clear correlation between throughput nor link stress, and the topology size and edge degree, it is evident that on average cross-layer algorithms outperform their simple overlay counterpart. For each algorithm, employing hierarchical clustering (HC) of the requesting peers consistently outperforms simple clustering based on network access link (SC). For mean transfer throughput, the augmented overlay algorithm with ASP provider selection exhibits significant gains over the rest of the algorithms, approaching 50% optimisation. It is also worth noting that for the 1000-node topology with a minimum edge degree of 4, all variants of the augmented algorithm provide mean link stress improvement on the order of 40%.

An interesting general observation from the simulation experiments is that the length of the content providers' list (i.e. the number of alternative sources) is not of major importance neither for increased transfer throughput nor for reduced link stress. On the other hand, providing partial and diverse views of this list to different peers based on network-local knowledge coupled with provider selection based on minimum AS hop distance (ASP), consistently provides for faster content replication as well as for improved network resource utilisation.

## V. Discussion and Conclusions

In this paper, we have proposed the concept of ISP hints as an elegant mean to provide application-network cross-layer optimization. ISP hints are obtained by an application through the explicit interaction with its access ISP, yet they do not contain any explicit information about the ISP's network structure or policies. We have shown through extensive performance evaluation that ISP hints provide synergistic optimisation benefiting both layers. The incentives for ISPs to deploy hint services are many fold. For instance, it allows them to reduce the impact of overlay applications on their network, and consequently improve the service delivered to all applications. Actually, it is conceivable that an ISP may choose to provide ISP hints to its customers to help it regain or maintain control over the traffic inside its network.

Indeed, this is because the choice of hints and their "values" can be made to fit the policies and strategies of the ISP and because clients whose ISP provides hints are unlikely to seek hints elsewhere which could jeopardize the local ISP's operations. Efficient ISP hints can be implemented from information readily available at the ISP. This makes hint services incrementally deployable. Furthermore, we have shown that hints can be effectively used as an integral part of the "signalling" and decision-making in applications. As a result, their effectiveness is independent of whether the applications encrypt the content or not, giving ISP hints some edge over approaches such as, for instance, caching. For the applications and corresponding overlay networks, the adoption of the use of hints is equally beneficial as these can dramatically improve perceived performance. This is the case not only in normal operational circumstances, but also in situations of extreme stress. Indeed, in the event of a flash crowd, for instance, and such situations have been shown to exists not only at fixed servers but also at overlay nodes [7][16], ISP hints can be very valuable in avoiding the bottleneck conditions, maintaining good levels of service and helping diffuse the situation. We therefore trust that there are enough incentives for all parties concerned to support and adopt the deployment of ISP hints. However, an ISP could provide further incentives to applications by affording better service to those that make efficient use of hints. Considering the example of P2P file distribution, an augmented region-aware overlay could receive more network resource on the basis that it tends to quickly "shed" load by redirecting requests outside of the ISP network. On the other hand, an ISP may want to try and enforce the use of hints by blocking or reducing the service of noncompliant applications.

In this paper, we have given some simple examples of ISP hints and how they can be used by applications. These are by no means restrictive: the possibilities opened by the deployment and use of ISP hints seem boundless.

## References


[1] Barabasi, A., L., Albert, R., Emergence of scaling in random networks, Science, pages 509–512, October 1999

[2] Bittorrent, http://bitconjurer.org/BitTorrent/

[3] Boston university representative internet topology generator (BRITE), http://www.cs.bu.edu/brite/

[4] Bu, T., Towsley, D., On distinguishing between Internet power law topology generators, IEEE INFOCOM'02, New York, USA, June 23-27, 2002

[5] Direct Connect, http://www.neo-modus.com/

[6] Gnutella, http://www.gnutella.com/

[7] Izal, M, Urvoy-Keller, G., Biersack, E., W., Felber, P.A., Al Hamra, A., Garces-Erice, L.,. Dissecting BitTorrent: Five Months in a Torrent's Lifetime, Passive and Active Measurement Workshop (PAM'04), April 19-20, 2004, Antibes Juan-les-Pins, France

[8] Karagiannis, T., Rodriguez, P., Papagiannaki, K., Should Internet service providers fear peer-assisted content distribution?, Internet Measurement Conference (IMC'05), October 19-21, 2005, Berkeley, CA, USA

[9] Kazaa media desktop, http://www.kazaa.com/

[10] Keralapura, C., C., R., Taft, N., Iannaconne, G., Can ISPs take the heat from overlay networks? In ACM Workshop on Hot Topics in Networks (HotNets'04), November 15-16, 2004, San Diego, CA, USA

[11] Nakao, A., Peterson, L., Bavier, A., A routing underlay for overlay networks, ACM SIGCOMM'03, August 25-29, 2003, Karlsruhe, Germany

[12] Ng, T., S., E., Zhang, H., A Network Positioning System for the Internet, in USENIX'04, Boston, MA, , June 27-July 2, 2004

[13] Ng, T., S., E., Zhang, H., Predicting Internet networking distance with coordinates-based approaches. In Proceedings of IEEE INFOCOM, June 2002

[14] Overnet/edonkey2000, http://www.edonkey2000.com/

[15] Pouwelse, J., A., Garbacki, P., Epema, D.,H.,J., Sips, H.,J., A measurement study of the bittorrent peer-to-peer file-sharing system, Delft University of Technology Parallel and Distributed Systems Report Series, Technical Report PDS-2004-007, 2004.

[16] Pouwelse, J., A., Garbacki, P., Epema, D.,H.,J., Sips, H.,J., The bittorrent p2p file-sharing system: measurements and analysis, the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05), February 24-25, 2005, Ithaca, NY, USA

[17] Rekhter,Y., Li, T., Hares, S., A Border Gateway Protocol 4 (BGP-4), IETF, Network Working Group, RFC4271, January 2006

[18] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/